EINDHOVEN UNIVERSITY OF TECHNOLOGY

Department of Mathematics and Computer Science

Examination Real-time Architectures (2IN60)
on Tuesday, January 13^{th} *2009, 9.00h-12.00h.*

First read the entire examination. There are 5 exercises in total. Grades are included between parentheses at all parts and sum up to 11 points. Good luck!

1. There exist three main classes of schedulability tests: *necessary*, *exact*, and *sufficient* tests.

   (a) (0.5) Give definitions for each of these classes of tests and explain the differences between them.

   (b) (0.5) Give an example for each of these classes of tests for fixed-priority preemptive scheduling (FPPS).

   (c) (1.0) Describe which conditions should be met in order to make the test applicable.

   Answers: See exercise 1 of the exam of 2IN25 of June 23^{rd}, 2005.

2. A sensor is used to detect the occurrence of a critical event E in the environment, which requires a reaction of the system within a finite amount of time D^E. As an example, a sensor could detect that a truck-bed is in the end position, and when the truck-bed enters the end position, the truck bed motor should not run longer than 400 ms. Assume a *polling* task τ to detect and handle the critical event.

   (a) (0.5) Draw a worst-case scenario for the critical event and the task τ.
   Answer: See RTA.Design page 14.

   (b) (0.5) Assume that the period T^τ and the deadline D^τ of task τ are equal, i.e. $T^\tau = D^\tau$. What would be appropriate values for T^τ? *Hint*: express T^τ in terms of D^E.
   Answer: $2T^\tau \leq D^E$; see RTA.Design page 14.

   (c) (0.5) When we lift the assumption that the period T^τ and the deadline D^τ are equal, what would be appropriate values for T^τ and D^τ? *Hint*: express T^τ and D^τ in terms of D^E.
   Answer: $T^\tau + D^\tau \leq D^E$; see RTA.Design page 14.

   (d) (0.5) Suppose we replace the sensor by another type of sensor that generates an interrupt which we want to handle by a task. What is the *kind* of the task that handles the interrupt (e.g. *periodic*, *sporadic*, or *asynchronous*), and what are the *parameters* characterizing the task? Motivate your answer!
   Answer: The exercise does not specify the minimum inter-arrival time of the interrupts. Hence, without loss of generality, we may classify the task as *a-periodic* having a deadline $D^\tau = D^E$. Unfortunately, it is impossible to determine the schedulability of a task set with such a hard rel-time task. As a result, we'd rather bound its period, e.g. to the reaction time D^E of the system, making it a *sporadic* task.

3. Give two task sets $\mathcal{T}_1$ and $\mathcal{T}_2$, where

   (a) (0.5) $\mathcal{T}_1$ is schedulable by fixed-priority pre-emptive scheduling (FPPS) but not by fixed-priority non-pre-emptive scheduling (FPNS);
   Answer: Compared to FPPS, FPNS can *increase* the worst-case response time of a higher priority task. Hence, by defining a task set $\mathcal{T}_1$ consisting of two tasks τ_1 and τ_2, where τ_1 has highest priority and $C_1 + C_2 > D_1$, we know that $\mathcal{T}$ is not schedulable under FPNS. Moreover, if $nC_1 + C_2 < \min(T_2, D_2, nT_1)$, then $\mathcal{T}_1$ is schedulable under FPPS.

   (b) (1.0) $\mathcal{T}_2$ is schedulable by FPNS but not by FPPS;
   Answer: See exercise 4 of the exam of 2IN25 of June 13^{th} 2008.

4. Consider a system consisting of a server, which is used to service aperiodic requests, and a set of hard real-time tasks. Describe the "response time analysis parameters" of the server for both best-case and worst-case response time analysis for the following types of servers:

   (a) (0.5) polling server;
   (b) (0.5) idling periodic server;
   (c) (0.5) gain-time providing periodic server;
   (d) (0.5) sporadic server;
   (e) (0.5) deferrable server.

   Answers: See RTA.Exercises-5 slide 8.

5. Consider a task set consisting of four hard real-time tasks τ_1, τ_2, τ_3, and τ_4, which share three resources R_1, R_2, and R_3. The tasks are scheduled using fixed priority scheduling, where task τ_i has a higher priority than τ_j iff $i < j$, i.e. τ_1 has highest and τ_4 has lowest priority. Figure 1 illustrates a situation of chained blocking when no resource sharing protocol is applied. Draw the timeline (including the dynamic priorities of the tasks) that results when applying the following resource sharing protocols for the example.

   (a) (1.0) Priority Inheritance Protocol (PIP).
   (b) (1.0) Priority Ceiling Protocol (PCP).
   (c) (1.0) Stack Resource Protocol (SRP).

   Answers: See RTA.Exercises-8.

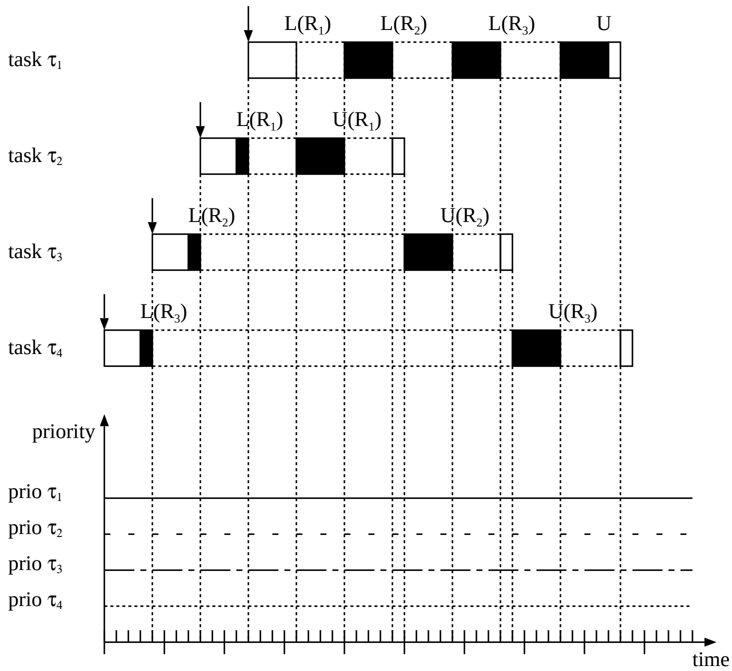

Figure 1: Example of chained blocking.